

Research Article

Energy Efficiency of Task Allocation for Embedded JPEG Systems

Yang-Hsin Fan,¹ Jan-Ou Wu,² and San-Fu Wang³

¹ Department of Computer Science and Information Engineering, National Taitung University, Taitung 95002, Taiwan

² Department of Electronic Engineering, De Lin Institute of Technology, New Taipei 23654, Taiwan

³ Department of Electronic Engineering, Ming Chi University of Technology, New Taipei 24301, Taiwan

Correspondence should be addressed to Yang-Hsin Fan; yhfan@nttu.edu.tw

Received 28 February 2014; Accepted 17 March 2014; Published 10 April 2014

Academic Editors: N. Barsoum, V. N. Dieu, and P. Vasant

Copyright © 2014 Yang-Hsin Fan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Embedded system works everywhere for repeatedly performing a few particular functionalities. Well-known products include consumer electronics, smart home applications, and telematics device, and so forth. Recently, developing methodology of embedded systems is applied to conduct the design of cloud embedded system resulting in the applications of embedded system being more diverse. However, the more energy consumes result from the more embedded system works. This study presents hyperrectangle technology (HT) to embedded system for obtaining energy saving. The HT adopts drift effect to construct embedded systems with more hardware circuits than software components or vice versa. It can fast construct embedded system with a set of hardware circuits and software components. Moreover, it has a great benefit to fast explore energy consumption for various embedded systems. The effects are presented by assessing a JPEG benchmarks. Experimental results demonstrate that the HT, respectively, achieves the energy saving by 29.84%, 2.07%, and 68.80% on average to GA, GHO, and Lin.

1. Introduction

Smarter, smaller, and portable characteristics make embedded systems to serve diverse functionalities. Nowadays, embedded systems have rapidly increasing requirements in the applications as in automobiles, avionics, and mobile devices. According to the IDC [1] reports, nearly 1 billion smart connected devices were shipped in 2011 and it would be double by 2016. Likewise, the Intel [2] predicts that there will be more than 30 billion devices constantly linked and another 150 billion fitfully connected in the end of this decade. It will greatly increase the demand of energy while these devices are served. However, the more embedded systems that are served, the more energy they consume. As a result, Bernd [3] summarizes the market and technology analysis which are towards energy efficiency for mobile devices, cloud computing, and storage services.

From an architectural perspective, all embedded systems have hardware and software components. These hardware components, such as *application specific integrated circuits*

(ASIC) or standard logic, offer specific functionalities or programmable gates when developing hardware circuits. Conversely, the software components, such as microcontrollers or programmable *digital signal processing* (DSPs), provide an environment for various application programs. By assembling these programmable hardware and software, any embedded system can be developed depending on system specifications.

Inside the embedded systems there are a few tasks that are designed by programmable hardware or software components. Each task consumes energy regardless of the forms in either programmable hardware or software components. Energy consumption is classified as dynamic and static energy consumption based on its mode of state. The dynamic energy consumption is defined while the task is working for providing functionalities. On the other hand, task consumes static energy dissipation when its state is idle. For an embedded system with n tasks, executing one task consumes dynamic energy and the other tasks arise static energy consumption. For example, when task 1 runs, task

2, task 3 till task n occur static energy consumption. In consideration of task 2 runs, it consumes dynamic energy consumption, at the same time, task 1, task 3, task 4 till task n occur static energy consumption. To iterate the process for every task execution, the energy consumption of embedded system can be assessed.

The aforementioned statements manifest embedded systems to incessantly consume energy for repeatedly performing a few particular tasks. In order to improve energy consumption, this study proposes *hyperrectangle technology* (HT) to embedded system target to obtain energy saving. This paper is organized as five sections. In Section 2, we investigate some previous works with respect to energy saving for embedded system. Section 3 describes HT for embedded systems to achieve energy saving. Section 4 demonstrates experimental results of *joint photographic experts group* (JPEG) encoding system that is performed by HT. We conclude this work in Section 5.

2. Preliminary Works

Researchers pay much attention to energy consumption of embedded systems with respect to the fields of processor energy consumption, real-time power consumption, dynamic power consumption, and scheduling power consumption. From the processors viewpoints, Vilcu [4] aims at real time embedded system to minimize the CPU power consumption. First, he studies task execution in the power consumption of processor(s). Then, he finds the effects of optimal configuration processor(s) for energy consumption. Finally, he defines globally optimal scheduling which gains minimal energy consumption for homogeneous multiprocessor system. Gao et al. [5] present *energy-efficient architecture for embedded software* (EAES) and dynamic energy-saving method to solve energy-saving problem. The former uses a processor with dynamic voltage scaling capability, FPGA modules, and extends directed acyclic graph to embedded system. The latter adopts preassignment to achieve dynamic runtime scheduling and minimize energy consumption. Qiu et al. [6] discuss the execution time of tasks with conditional instructions or operations problem. They adopt probabilistic random variable approach to model execution time of tasks. Then, they propose practical algorithm VACP to minimize energy consumption for uniprocessor embedded systems. Silva-Filho and Lima [7] state that memory hierarchy consumes power up to 50% in microprocessor system. Consequently, they propose an automated architecture exploration mechanism to NIOS II processor and memory hierarchy with parameter variation. The experimental results show the reduction of energy consumption is approximately 27%. In 2008, Zeng et al. [8] present generalized *dynamic energy performance scaling* (DEPS) framework to hard real-time embedded systems for exploring application-specific energy-saving potential issue. Three energy performance tradeoff technologies, DHRC, DVFS, and DPM, are integrated into DEPS. Their experimental results of simulation show the static DEPS has been improved, respectively, 13.6% and 13.7% in DVFS and DHRC. Also, dynamic DEPS improves 5.7% when comparing to static DEPS.

Real-time power information is a valuable data for software designer for battery-powered embedded systems. Genser et al. [9] propose power profiling approach to collect real-time power information at early designing stages. Moreover, they present an emulation-based power profiling approach to achieve real-time power analysis for embedded systems. Because the power information is collected at early designing stages, the development efficiency and time to market are improved. In 2008, Elewi et al. [10] first discuss the real-time scheduling of dependent tasks problem and then present enhanced *multispeed* (MS) algorithm for energy saving. With energy consumption problem of battery-powered embedded systems, Casares et al. [11] aim at embedded smart camera to analyze the power consumption and performance. Not only graph of energy consumption but also instructions of collections are presented. They conclude the importance of lightweight algorithm, the time of transfer data, and transferred data type.

Dynamic power consumption of *field programmable gate array* (FPGA) is discussed in [12, 13]. In 2009, Tsang and So [12] adopt precomputation approach to reduce dynamic power consumption in commercial off-the-shelf FPGAs. The experimental results of comparator show that 83% of dynamic power in logic or 43.1% of total dynamic power is reduced if the increased resource consumption can be negligible. In 2010, Bhandari et al. [13] present fly partial reconfiguration as well as scaling the clock on FPGA for reducing dynamic power consumption in embedded system. They conclude that the factors of dynamic power dissipation consist of application, architecture, and reconfigurable time.

The application of different algorithms to arrange scheduling issues for reducing power consumption is discussed in [14, 15]. In 2010, Bashiri and Miremadi [14] investigate *earliest-deadline-first* (EDF) and *rate-monotonic* (RM) algorithms on power efficiency of task scheduling. The results show that the BF-EDF and FF-EDF have the best power efficiency. In 2011, Cho et al. [15] propose power-saving scheduling algorithm and use soft-deadline to reduce energy consumption by about 40%. However, it is a trade off on performance and power savings for embedded systems. Kan et al. [16] present a heuristic algorithm called TGPM-ALL with interior point method to handle the frequency assignment on multiple soft-deadlines embedded systems. Their empirical results show the effectiveness in comparing TGPM-ALL with TGPM-I and BEST algorithms.

3. Hyperrectangle Approach

Task graph (TG) is a conceptual graph that facilitates to describe operation for embedded system. TG comprises of vertices (V), edges (A), and levels (L) that can be represented by a 3-tuple set, $G(V, A, L)$. The V is a unit of work which may take dependencies one or more antecedents. The A is used to exhibit the flow among V . The L indicates the order of works for V . Based on TG, we propose *energy-consumed task graph* (ETG) as system model of HT that adds a factor of energy consumption on TG.

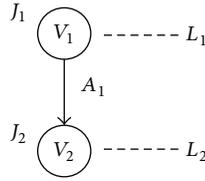


FIGURE 1: A simple ETG with two tasks.

3.1. System Model. ETG comprises of vertices (V), edges (A), levels (L), and energy consumption (J) that can be represented by a 4-tuple set, $G(V, A, L, J)$. Symbol V stands for task that is a working unit on embedded system. For example, one task is represented as V_1 and a number of tasks are labeled as V_2, V_3 , and V_4 , and so forth. Hence, application program inside embedded system can be defined as a set of tasks as V_1, V_2 , and V_3 to V_n . Another symbol A is used to direct work flow of applications among V s. For instance, symbol A_1 guides the working flow from task V_1 to another task V_2 . Moreover, two tasks are connected by A_1 that implies their location on different levels. Label L defines the state of V and the height of ETG. The state is organized into two categories. One state is named working (W) and the other idle (I). Both states simultaneously incur when a symbol A_1 activates. For instance, working state W_1 on V_2 and idle state I_1 on V_1 separately form when A_1 activates. Sign J denotes the energy consumption of V . Each task consumes energy depending on either W or I . Figure 1 displays a ETG with 2 vertices (i.e., V_1 and V_2), 1 edge, 2 energy consumption (i.e., J_1 and J_2), and 2 levels (i.e., L_1 and L_2).

3.2. Energy Consumption Definition. Power consumption of the task is classified into dynamic or static power consumption according to their state of work. Dynamic power consumption D occurs while the task locates W . On the contrary, the task in I consumes static power consumption S . Take Figure 1 as an example, the V_1 and V_2 first consume D and S separately in L_1 because the former locates at W and the latter places on I . After that, the A_1 directs the work flow to L_2 . In L_2 , the V_2 and V_1 individually consume D on W and S on I . In summary, both D and S are consumed by V depending on W or I . It should be noticed that each task must consume energy at any time even though it is idle.

Low power dissipation model and analysis for embedded systems are discussed by Fan et al. [17]. They derive power dissipation with dynamic and static power dissipation from TG. The expression of the sum of power consumption for embedded system is calculated by using

$$P = (L - 1) \times (P_{s,t_1} + P_{s,t_2} + \dots + P_{s,t_n}) + (P_{d,t_1} + P_{d,t_2} + \dots + P_{d,t_n}), \quad (1)$$

where L is the height of TG, P_s is static power consumption, P_d is dynamic power consumption, and t_1, t_2, \dots, t_n is a set of task. In consideration of energy consumption, the energy consumption is formulated as follows:

$$E = P \times T, \quad (2)$$

where P is power dissipation and T represents execution time. Owing to power consumption which is divided into D and S , the energy consumption is categorized into dynamic (E_d) and static (E_s) energy dissipation. Moreover, each task can be separately implemented as two forms of f as *hardware circuit* (HC) and *software component* (SC) so that the energy consumption of embedded system can be formulated as follows:

$$E_{d,v_i}^f = P_{d,v_i}^f \times T_{d,v_i}^f, \quad (3)$$

$$E_{s,v_i}^f = P_{s,v_i}^f \times T_{s,v_i}^f,$$

where d is dynamic energy consumption, s is static energy consumption, f is a form of hardware circuit or software component, and v is task, $i = 1, 2, \dots, n$. In summarizing equations from (1) to (3), the total energy consumption can be derived as follows:

$$E = (L - 1) \times (E_{s,v_1}^f + E_{s,v_2}^f + \dots + E_{s,v_n}^f) + (E_{d,v_1}^f + E_{d,v_2}^f + \dots + E_{d,v_n}^f), \quad (4)$$

where L is the height of ETG, E_s is static energy consumption, and E_d is dynamic energy consumption.

3.3. Hyperrectangle Model. To construct hyperrectangle model of energy consumption for embedded system, we first analyze the energy consumption of ETG with two tasks which is shown in Figure 1. Then derive complicated model from it. By holding the principle of one task that has two forms (i.e., HC and SC), a ETG with two tasks (i.e., v_1 and v_2) can be constructed four embedded systems namely HC-HC (E_1), HC-SC (E_2), SC-HC (E_3), and SC-SC (E_4). According to (3), the energy consumption of E_1, E_2, E_3 , and E_4 can be defined in the following:

$$E_1 = E_{d,v_1}^{HC} + E_{s,v_2}^{HC} + E_{d,v_2}^{HC} + E_{s,v_1}^{HC}, \quad (5)$$

$$E_2 = E_{d,v_1}^{HC} + E_{s,v_2}^{SC} + E_{d,v_2}^{SC} + E_{s,v_1}^{HC}, \quad (6)$$

$$E_3 = E_{d,v_1}^{SC} + E_{s,v_2}^{HC} + E_{d,v_2}^{HC} + E_{s,v_1}^{SC}, \quad (7)$$

$$E_4 = E_{d,v_1}^{SC} + E_{s,v_2}^{SC} + E_{d,v_2}^{SC} + E_{s,v_1}^{SC}. \quad (8)$$

The first and the second terms or the third and the fourth terms from (5) to (8) prove the description in Section 3.1, which indicates that W and I simultaneously incur while a symbol A_1 activates. Moreover, the third and the fourth terms can be regarded as mutual functions as the second and the first terms. Consequently, the third and the fourth terms from (5) to (8) can be formulated as the third term in the following:

$$E_1 = E_{d,v_1}^{HC} + E_{s,v_2}^{HC} + f(E_{d,v_1}^{HC}, E_{s,v_2}^{HC}), \quad (9)$$

$$E_2 = E_{d,v_1}^{HC} + E_{s,v_2}^{SC} + f(E_{d,v_1}^{HC}, E_{s,v_2}^{SC}), \quad (10)$$

$$E_3 = E_{d,v_1}^{SC} + E_{s,v_2}^{HC} + f(E_{d,v_1}^{SC}, E_{s,v_2}^{HC}), \quad (11)$$

$$E_4 = E_{d,v_1}^{SC} + E_{s,v_2}^{SC} + f(E_{d,v_1}^{SC}, E_{s,v_2}^{SC}). \quad (12)$$

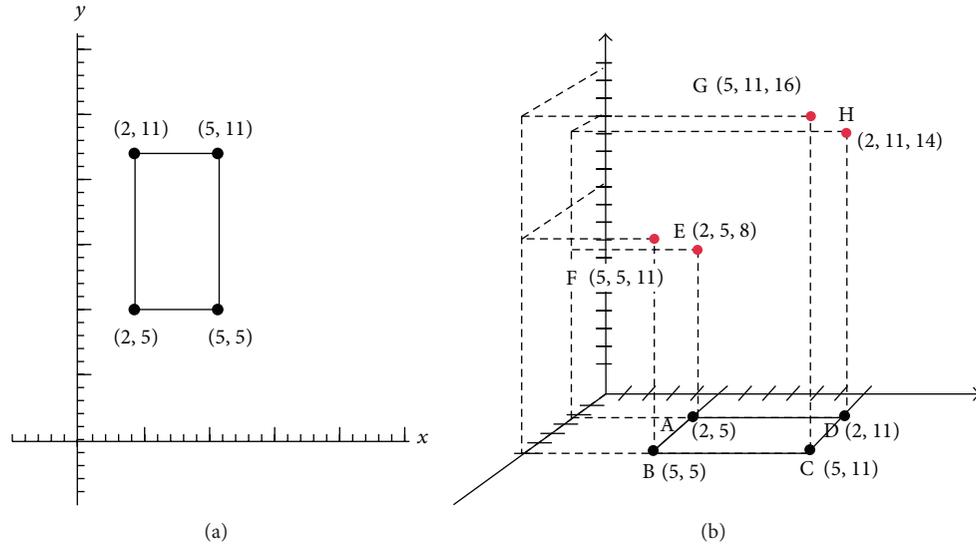


FIGURE 2: Example of a rectangle R^2 in hyperrectangle model.

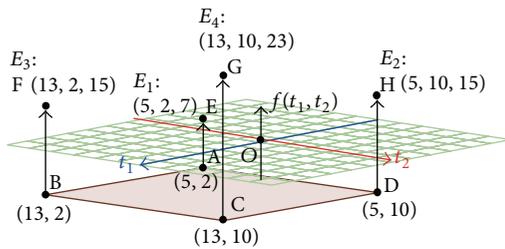


FIGURE 3: Hyperrectangle schema of embedded system with two tasks.

From the axial coordination's perspective, the first and the second terms from (9) to (12) can be represented as four points of a rectangle R^1 . Moreover, the third term from (9) to (12) forms four points in another rectangle R^2 in the tridimensionality. As a result, the hyperrectangle model of energy consumption for embedded system with two tasks is constructed and transferred to the three-dimensional space.

Based on the previous description, the hyperrectangle model of energy consumption for embedded system with two tasks can be defined in the following.

For a rectangle R^2 , it has vertex (x_i, y_j) , $i, j = 1, 2$, and a solution set $D \subseteq R^2$. A given function $f: D \rightarrow R$, $f(x, y) = u$ and $b \in R$, the solution can be obtained as follows:

$$x_i + y_j + f(x_i, y_j) \leq b, \quad \text{where } i, j = 1, 2. \quad (13)$$

Example 1. Figure 2(a) shows a set of vertex $D = \{(2, 5), (5, 5), (2, 11), (5, 11)\}$. Figure 2(b) displays $f: D \rightarrow R$ where $f(2, 5) = 8$, $f(5, 5) = 11$, $f(2, 11) = 14$, and $f(5, 11) = 16$. For a given $b = 20$, the solution comprises $\{(2, 5), (5, 5)\}$ since it meets $x_i + y_j + f(x_i, y_j) \leq b$.

Example 2. Table 1 displays an energy consumption example of embedded system with two tasks. The evaluating factors of

TABLE 1: Energy consumption of embedded system with 2 tasks.

Tasks	Energy consumption			
	HC	SC	HC	SC
	E_d^{HC}	E_s^{HW}	E_d^{SC}	E_s^{SC}
v_1	5	1	13	9
v_2	6	2	14	10

energy consumption include the name of task, dynamic, and static energy consumption of HC and SC. Figure 3 shows a hyperrectangle schema of embedded system with two tasks (i.e., v_1 and v_2). According to (9) to (12), the first and the second term are transferred to XY axes for E_1, E_2, E_3 , and E_4 , where locates at points A, D, B, and C. Points E, H, F, and G are marked for the third term from (9) to (12). We observe the most energy consumption occurring at point G (i.e., $13 + 10 + 23 = 46$), which comprises SC-SC (E_4). On the other hand, the HC-HC (E_1) consumes the fewest energy consumption. Eventually, the energy consumption can be improved if one task is substituted from SC to HC where less energy is consumed.

Similarly, an embedded system with three tasks can be defined as follows. For a cuboid R^3 , it has vertex (x_i, y_j, z_k) , $i, j, k = 1, 2$, and a solution set $D \subseteq R^3$. A given function $f: D \rightarrow R$, $f(x, y, z) = v$ and $b \in R$, the solution can be obtained from the following:

$$x_i + y_j + z_k + f(x_i, y_j, z_k) \leq b, \quad \text{where } i, j, k = 1, 2. \quad (14)$$

According to (13) and (14), we summarize the general expression for embedded system with n tasks as follows. For a hyperrectangle R^n , it has vertex, $(x_{(1,i_1)}, x_{(2,i_2)}, \dots, x_{(n,i_n)})$, $i_j = 1, 2$, and a solution set $D \subseteq R^n$ where D is shown in

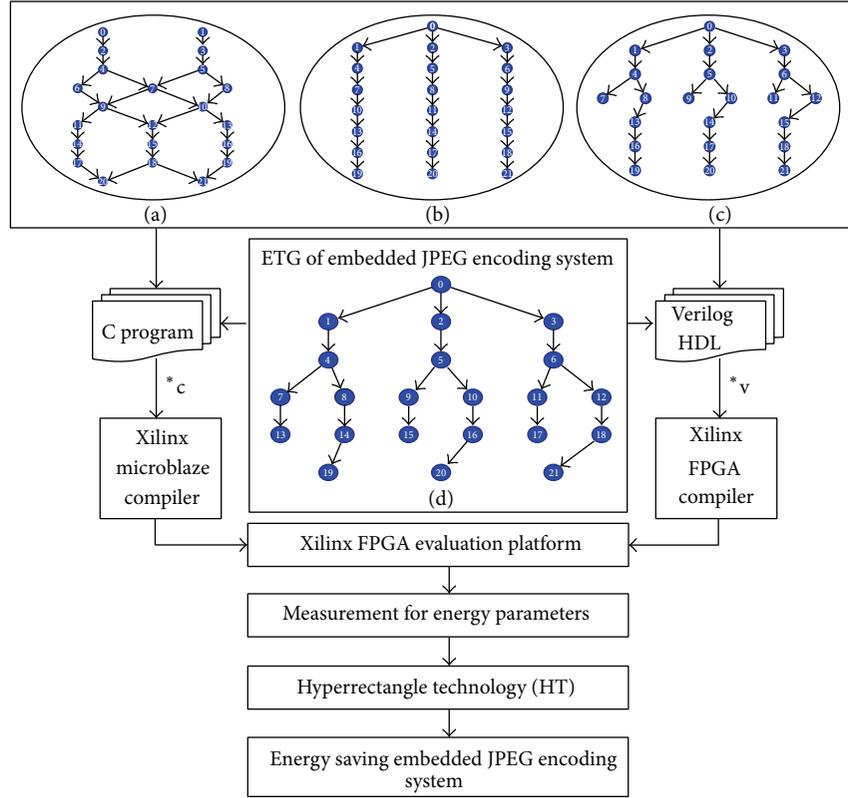


FIGURE 4: Flow chart of experimental setup.

TABLE 2: Technology of system parameters.

	GA	GHO
Number of chromosomes	100000	100000
Number of populations	500	500
Probability of crossover	1	1
Probability of mutation	1	1

(15). Given a function $f : D \rightarrow R$ and $b \in R$, the solution can be obtained as follows:

$$D = \left\{ \left(x_{(1,i_1)}, x_{(2,i_2)}, \dots, x_{(n,i_n)} \right) \mid i_j = 1, 2 \right\}, \quad (15)$$

$$x_{(1,i_1)} + x_{(2,i_2)} + \dots + x_{(n,i_n)} + f \left(x_{(1,i_1)}, x_{(2,i_2)}, \dots, x_{(n,i_n)} \right) \leq b, \quad (16)$$

where $1 \leq j \leq n$.

Applying hyperrectangle approach to embedded systems for gaining energy consumption consists of the following steps. First, the number of tasks U of embedded system must be defined. Next, separately constructing the number of tasks $U_{hc,i}$ with HC and $U_{sc,j}$ with SC being the same as U . Third, constructing sets of $f(U_{hc,i})$ and $f(U_{sc,j})$ depicts energy consumption for each task. Therefore, energy consumption of each task can be fast evaluated. Fourth, constructing the first embedded system ES_a that comprises a set of tasks with U_{sc} . Fifth, the task with the most energy consumption in $f(U_{sc,a})$ is swapped with $f(U_{hc,a})$. After the swapping process

is iterated until each task is made of U_{hc} , the HT exploits a set of embedded systems HT_x according to the number of U_{hc} and U_{sc} . The first embedded system ES_1 is assembled by one of two U_{hc} and U_{sc} . If U is odd, extra SC is set preceding privilege to deploy to the ES_1 . The second embedded system ES_2 is set where the number of tasks with SC is more than HC. Alternatively, the number of tasks with HC that is more than SC is set to the third embedded system ES_3 . In setting ES_2 and ES_3 until ES_m , the above process is executed repeatedly. We observe that ES_{2i} and $ES_{(2i+1)}$ form the drift effect with SC and HC, respectively.

4. Experimental Results and Analysis

The experimental platform is Xilinx FPGA ML507 [18]. Table 2 presents the technology of system parameters. The tested example is *joint photographic experts group* (JPEG) encoding system that consists of 22 tasks and 9 levels. From level 1 to 9 in Figure 4(a), the number of tasks is 2, 2, 2, 3, 2, 3, 3, 3, and 2, respectively. Each task is individually implemented as hardware circuit and software component form, which are designed by Verilog programming language and C programming language. Figure 4 demonstrates the flow chart of experimental setup.

The measured data of energy consumption is shown in Table 3. In the Task column, it shows the name of task that works in the JPEG encoding system. For the dynamic and static energy consumption of hardware circuit, it is displayed

TABLE 3: Measured data of tasks for JPEG encoding system.

Tasks	Energy Consumption			
	E_d^{HC}	HC (10^{-6})	E_s^{HC}	SC (10^{-3})
v_0 (LevelOffset)	0.0180504		0.0119968	0.591286362
v_1 (DCT)	1.414320718		0.631151123	181.6586949
v_2 (DCT)	1.414320718		0.631151123	181.6586949
v_3 (DCT)	1.414320718		0.631151123	181.6586949
v_4 (Quant.)	0.629914608		0.123049296	0.93692231
v_5 (Quant.)	0.629914608		0.123049296	0.93692231
v_6 (Quant.)	0.629914608		0.123049296	0.93692231
v_7 (DPCM)	0.000575667		0.000336501	0.040158143
v_8 (ZigZag)	0.02977856		0.003429459	0.561685068
v_9 (DPCM)	0.000575667		0.000336501	0.040158143
v_{10} (ZigZag)	0.02977856		0.003429459	0.561685068
v_{11} (DPCM)	0.000575667		0.000336501	0.040158143
v_{12} (ZigZag)	0.02977856		0.003429459	0.561685068
v_{13} (VLC)	0.086972024		0.028369015	0.054136441
v_{14} (RLE)	0.046884314		0.023175988	0.838115122
v_{15} (VLC)	0.093019754		0.030341698	0.054136441
v_{16} (RLE)	0.046884314		0.023175988	0.838115122
v_{17} (VLC)	0.093019754		0.030341698	0.054136441
v_{18} (RLE)	0.046884314		0.023175988	0.838115122
v_{19} (VLC)	0.116959532		0.022825184	0.996227367
v_{20} (VLC)	0.116959532		0.022825184	0.976457285
v_{21} (VLC)	0.116959532		0.022825184	0.976457285

TABLE 4: Energy consumption of HT.

Embedded systems	HT (ours)				
	v_1	v_{22}	Energy dissipation (J)		
		Structure 1: Figure 4(a)	Structure 2: Figure 4(b)	Structure 3: Figure 4(c)	Structure 4: Figure 4(d)
$ES_a(00000000000000000000)$		3.9713661	3.544421533	3.117476966	2.69053
$ES_b(01000000000000000000)$		2.680031235	2.391796366	2.103561497	1.81533
$ES_c(01100000000000000000)$		1.38869637	1.239171199	1.089646028	0.94012
$ES_d(01110000000000000000)$		0.097361505	0.086546032	0.075730559	0.06492
$ES_e(011100000000000000100)$		0.088413403	0.078591929	0.068770455	0.05895
$ES_f(011100000000000000110)$		0.079642539	0.070795381	0.061948222	0.05310
$ES_g(011100000000000000111)$		0.070871676	0.062998833	0.05512599	0.04725
$ES_h(0111100000000000000111)$		0.0624566	0.055518605	0.04858061	0.04164
$ES_i(0111110000000000000111)$		0.054041524	0.048038377	0.04203523	0.03603
$ES_j(0111111000000000000111)$		0.045626448	0.040558149	0.03548985	0.03042
$ES_k(0111111100000010000111)$		0.038096141	0.033864372	0.029632602	0.02540
$ES_l(0111111100000010100111)$		0.030565834	0.027170595	0.023775355	0.02038
$ES_m(0111111100000010101111)$		0.023035527	0.020476818	0.017918108	0.01536
$ES_n(1111111000000010101111)$		0.017720291	0.015752077	0.013783864	0.01182
$ES_o(1111111101000010101111)$		0.012670802	0.011263568	0.009856333	0.00845
$ES_p(11111111010100010101111)$		0.007621314	0.006775058	0.005928803	0.00508
$ES_q(11111111010101010101111)$		0.002571825	0.002286549	0.002001273	0.00172
$ES_r(11111111010101110101111)$		0.002084963	0.001853789	0.001622614	0.00139
$ES_s(11111111010101111101111)$		0.001598124	0.001421048	0.001243973	0.00107
$ES_t(11111111010101111111111)$		0.001111284	0.000988308	0.000865333	0.00074
$ES_u(1111111110101111111111)$		0.000749892	0.000667071	0.00058425	0.00050
$ES_v(1111111111011111111111)$		0.000388501	0.000345834	0.000303167	0.00026
$ES_w(1111111111111111111111)$		2.711E - 05	2.4597E - 05	2.20841E - 05	1.95711E - 05

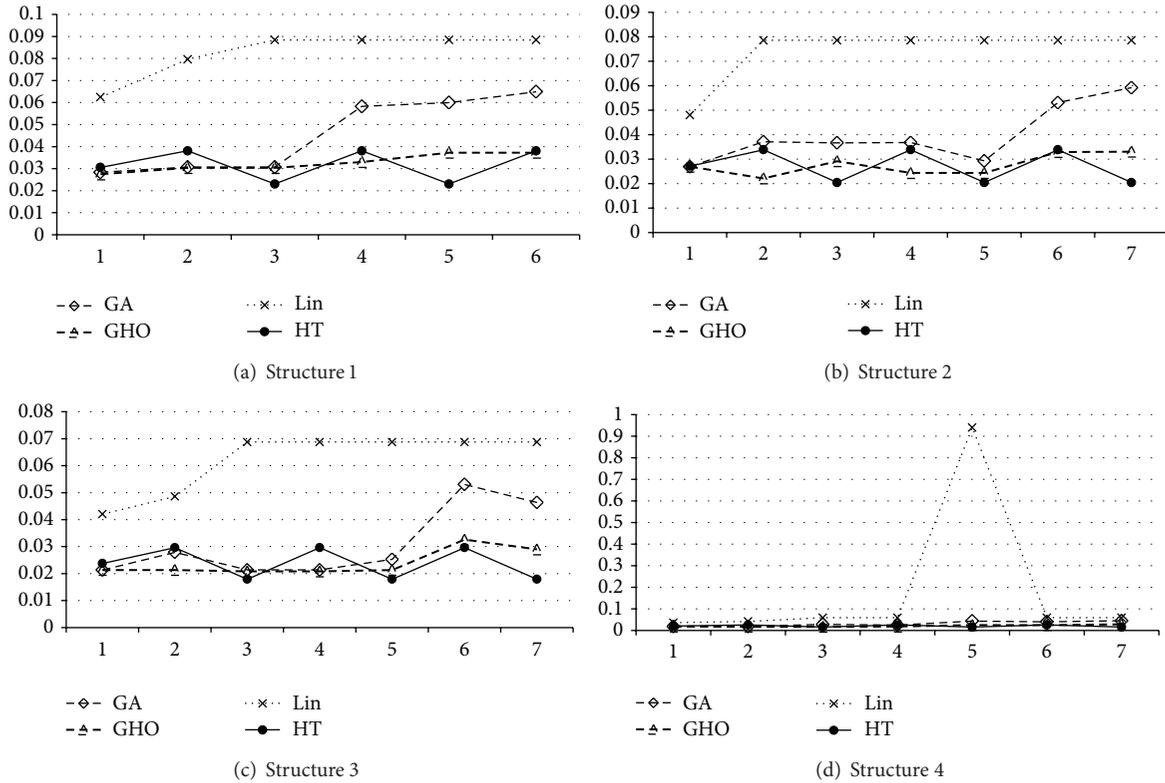


FIGURE 5: Energy consumption of embedded JPEG encoding system.

in column 2 and 3. The software tasks with dynamic and static energy consumption are illustrated in column 4 and 5.

Table 4 lists the experimental results of the proposed approach. Embedded systems column displays the results according to (16). In the energy dissipation column, it is calculated via (4). The ES_1 is set to ES_1 , ES_k , and ES_m is set to ES_2 and ES_3 , respectively. On one hand, the SC drift effect, respectively, diffuses from ES_k to ES_a . Similarly, the HC drift effect diffuses from ES_m to ES_w , respectively. All designs by HT to embedded systems can be fast explored for energy consumption.

To present the efficiency of the proposed HT, we compare HT to *genetic algorithm* (GA) [19], GHO [20], and Lin [21] via JPEG benchmarks. Four structures shown in Figure 4 of energy consumption are used to demonstrate the effects of HT. The energy consumption of each structure is set to 0.1, 0.09, 0.08, and 1 joule, respectively. The structure 1 is shown in Figure 4(a) and the experimental result is depicted in Figure 5(a). The HT gains the energy saving in comparison with GA [19], GHO [20], and Lin [21]. Moreover, the HT improves the energy consumption by 30.00%, 2.38%, and 61.49% on average to GA [19], GHO [20], and Lin [21], respectively. Figure 5(b) displays the results of structure 2 (Figure 4(b)), in which the HT separately improves the energy consumption by 31.89%, 1.52%, and 63.40% on average to GA [19], GHO [20], and Lin [21]. Figure 5(c) shows the results of structure 3 (Figure 4(c)), in which the HT individually improves the energy consumption by 23.18%, 0.44%

and, 61.69% on average to GA [19], GHO [20], and Lin [21]. Figure 5(d) exhibits the results of structure 4 (Figure 4(d)), in which the HT separately improves the energy consumption by 34.28%, 3.93%, and 88.62% on average to GA [19], GHO [20], and Lin [21]. In summary, the HT achieves the energy saving by 29.84%, 2.07%, and 68.80% on average to GA [19], GHO [20], and Lin [21], respectively.

5. Conclusion

Energy saving issue is always discussed and concerned in electronic devices. Nowadays, nearly any electronic device either already has existed or will embed computing systems resulting in the applications of embedded systems that are more diverse. It reveals that embedded systems are growing exponentially. While more and more embedded systems are repeated day by day in order to provide various functionalities, the speed of energy consumption is greatly increased.

This study presents *hyperrectangle technology* (HT) to embedded systems target to achieve energy saving. The drift effect on HT facilitates the designer to fast explore energy consumption of embedded systems. The effectiveness of the proposed approach is demonstrated by assessing a JPEG benchmarks. Experimental results demonstrate that the HT achieves the energy saving by 29.84%, 2.07%, and 68.80% on average to GA, GHO, and Lin, respectively. Consequently, this work is valuable for developing energy-saving embedded systems.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgment

The authors would like to thank the National Science Council, Taiwan, for financially supporting this research under Contract nos. NSC 101-2221-E-143-002 and NSC 102-2221-E-143-002.

References

- [1] Businesswire web site, http://www.Businesswire.com/news/home/20120328005370/en/1-Billion-Smart-Connected-Devices-Shipped-2011#.U0Qsr_mSx8E.
- [2] Intel web site, February 2013, <http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/consumerization-of-it-trends-whitepaper.pdf>.
- [3] P. S. Bernd, "Technology, market and cost trends 2012," CERN IT Report, 2012, https://espace.cern.ch/WLCG-document-repository/Technical_Documents/Technology_Market_Cost_Trends_2012_v23.pdf.
- [4] D. Vilcu, "Real time scheduling and CPU power consumption in embedded systems," in *Proceedings of the 2008 IEEE International Conference on Automation, Quality and Testing, Robotics*, pp. 261–266, May 2008.
- [5] Z. Gao, G. Dai, P. Liu, and P. Zhang, "Energy-efficient architecture for embedded software with hard real-time requirements in partial reconfigurable systems," in *Proceedings of the 2009 IEEE International Conference on Embedded Computing, Scalable Computing, and Communications*, pp. 387–392, September 2009.
- [6] M. Qiu, J. Wu, F. Hu, S. Liu, and L. Wang, "Voltage assignment for soft real-time embedded systems with continuous probability distribution," in *Proceedings of the 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 413–418, August 2009.
- [7] A. G. Silva-Filho and S. M. L. Lima, "Energy consumption reduction mechanism by tuning cache configuration using nios II processor," in *Proceedings of the 2008 IEEE International SOC Conference*, pp. 291–294, September 2008.
- [8] G. Zeng, H. Tomiyama, H. Takada, and T. Ishihara, "A generalized framework for system-wide energy savings in hard real-time embedded systems," in *Proceedings of the IEEE/IFIP International Conference on Embedded and Ubiquitous Computing (EUC '08)*, pp. 206–213, December 2008.
- [9] A. Genser, C. Bachmann, J. Haid, C. Steger, and R. Weiss, "An emulation-based real-time power profiling unit for embedded software," in *Proceedings of the IEEE International Symposium on Systems, Architectures, Modelling, and Simulation*, pp. 67–73, July 2009.
- [10] A. M. Elewi, M. H. A. Awadalla, and M. I. Eladawy, "Energy-efficient multi-speed algorithm for scheduling dependent real-time tasks," in *Proceedings of the International Conference on Computer Engineering and Systems (ICCES '08)*, pp. 237–242, November 2008.
- [11] M. Casares, A. Pinto, Y. Wang, and S. Velipasalar, "Power consumption and performance analysis of object tracking and event detection with wireless embedded smart cameras," in *Proceedings of the 3rd International Conference on Signal Processing and Communication Systems (ICSPCS '09)*, pp. 1–8, September 2009.
- [12] C. C. Tsang and H. K.-H. So, "Reducing dynamic power consumption in FPGAs using precomputation," in *Proceedings of the IEEE International Conference on Field-Programmable Technology (FPT '09)*, pp. 407–410, December 2009.
- [13] S. U. Bhandari, S. Subbaraman, and S. Pujari, "Power reduction in embedded system on FPGA using on the fly partial reconfiguration," in *Proceedings of the 2010 IEEE International Symposium on Electronic System Design*, pp. 77–80, December 2010.
- [14] M. Bashiri and S. G. Miremadi, "Investigating the effects of schedulability conditions on the power efficiency of task scheduling in an embedded system," in *Proceedings of the 13th IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing*, pp. 102–106, May 2010.
- [15] K. Cho, C. Liang, J. Huang, and C. Yang, "Design and implementation of a general purpose power-saving scheduling algorithm for embedded systems," in *Proceedings of the IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC '11)*, pp. 1–5, September 2011.
- [16] E. Y. Y. Kan, W. K. Chan, and T. H. Tse, "Leveraging performance and power savings for embedded systems using multiple target deadlines," in *Proceedings of the IEEE International Conference on Quality Software*, pp. 473–480, July 2010.
- [17] Y.-H. Fan, J.-O. Wu, and S.-F. Wang, "Low power dissipation model analysis for embedded systems," *Journal of Research Notes in Information Science*, vol. 13, pp. 184–188, 2013.
- [18] Xilinx ML507 website, <http://www.xilinx.com>.
- [19] Y. Zou, Z. Zhuang, and H. Chen, "HW-SW partitioning based on genetic algorithm," in *Proceedings of the 2004 Congress on Evolutionary Computation (CEC '04)*, pp. 628–633, Portland, Ore, USA, June 2004.
- [20] T.-Y. Lee, Y.-H. Fan, Y.-M. Cheng, and C.-C. Tsai, "Hardware-software partitioning for embedded multiprocessor FPGA systems," *Journal of Innovative Computing, Information and Control A*, vol. 5, no. 10, pp. 3071–3083, 2009.
- [21] T. Y. Lin, Y. T. Hung, and R. G. Chang, "Efficient hardware/software partitioning approach for embedded multiprocessor systems," in *Proceedings of the International Symposium on VLSI Design, Automation and Test (VLSI-DAT '06)*, pp. 231–234, Hsinchu, Taiwan, April 2007.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

